EEEEEEEEEEEEE	RRRRRRRRRRRR	FFFFFFFFFFFFF
EEEEEEEEEEEEE	RRRRRRRRRRR	FFFFFFFFFFFFF
ÉÉÉÉÉÉÉÉÉÉÉÉÉÉ	RRRRRRRRRRR	FFFFFFFFFFFFF
EEE	RRR RRR	FFF
EEE		
	RRR RRR	FFF
EEE	RRR RRR	FFF
EEE	RRR RRR	FFF
EEE	RRR RRR	FFF
ĒĒĒ	RRR RRR	FFF
EEEEEEEEEE	RRRRRRRRRRR	FFFFFFFFFF
EEEEEEEEEEE	RRRRRRRRRRRR	FFFFFFFFFF
EEEEEEEEEE	RRRRRRRRRRRR	FFFFFFFFFF
EEE	RRR RRR	FFF
	******	
EEE	RRR RRR	FFF
EEEEEEEEEEEE	RRR RRR	FFF
EEEEEEEEEEEE	RRR RRR	FFF
EEEEEEEEEEEE	RRR RRR	FFF
	mm nm	111

77777777888888888888899999999999999999
--

DUP

DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD	PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP	11 1111 1111 1111 111 111 111 111111	11 1111 1111 1111 11 11 11 11 11 11 1111	••••
	\$			

(++

```
SUBROUTINE DUP11 (lun,parcsr,rxcsr,rxdbuf,txcsr,txdbuf)
```

Version: 'V04-000'

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICFISE AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

Author: Sharon Reynolds

Creation date: 8-Sept-80

functional description:

This module decodes the Dup11 register contents. It is called by the DUP3271 module.

Modified by:

V03-002 SAR0112 Sharon A. Reynolds, 23-Jun-1983 Changed the carriage control in the 'format' statements for use with ERF.

v03-001 BP0001 Brian Porter, 20-AUG-1982 Minor edit.

Byte

lun

Integer\*4 addrs
Integer\*4 char
Integer\*4 field

DUF

Page

```
0058
0059
                         Integer*4
                                                 temp
                         Integer*4
                                                 rxcsr
0060
0061
0062
0063
0064
0065
                         Integer*4
                                                 rxdbuf
                         Integer#4
                                                 txcsr
                         Integer*4
                                                 txdbuf
                         Integer * 4
                                                 parcsr
                         Integer*4
                                                 compressC
                         Integer*4
                                                 compress4
0066
0067
                         logical*1
                                                 diagnostic_mode
0068
0069
0070
                                                protocol(0:1)
oner_mode(0:1)
rxcsr_1(0:0)
rxcsr_2(3:8)
rxcsr_3(10:11)
rxcsr_4(14:15)
rxdbuf_1(8:10)
rxdbuf_3(14:15)
txcsr_1(4:4)
txcsr_2(6:9)
txcsr_3(15:15)
txdbuf_1(8:10)
parcsr_1(9:9)
dtr_status(0:1)
rts_status(0:1)
car_status(0:1)
cts_status(0:1)
                         Character*23
                                                 protocol(0:1)
                        Character*12
0071
                         Character*18
0072
0073
                         Character*26
                         Character*24
0074
                         Character + 18
0075
                         Character*26
0076
                         Character * 19
0077
                         Character*17
0078
                         Character*5
0079
                         Character*29
0080
                         Character*28
0081
                        Character*27
0082
                         Character + 12
0083
                         Character * 26
0084
                         Character*22
0085
                         Character + 21
0086
0087
                         Character*21
                        Character*20
0088
0089
0090
0091
            C
0092
                        Define text for bits in the RXCSR register
0093
0094
                                    rxcsr_1(0)
rxcsr_2(3)
rxcsr_2(4)
rxcsr_2(5)
rxcsr_2(6)
rxcsr_2(7)
rxcsr_2(8)
rxcsr_3(10)
rxcsr_3(11)
rxcsr_4(14)
rxcsr_4(15)
0095
                                                             /'DATA SET CHANGE B+'/
                        Data
0096
                                                             /'SECONDARY TRANSMIT DATA+'/
                        Data
                                                            /'RECEIVER ENABLE*'/
/'DATA SET INTERRUPT ENABLE*'/
/'RECEIVER INTERRUPT ENABLE*'/
0097
                        Data
0098
                        Data
0099
                        Data
0100
                                                             /'RECEIVER DONE+'/
                        Data
                                                             /'STRIP SYNC CHARACTER+'/
/'SECONDARY RECEIVED DATA+'/
0101
                        Data
0102
                        Data
                                                             /'RECEIVER ACTIVE+'/
0103
                        Data
0104
                                                             /'RING+'/
                        Data
0105
                        Data
                                                             /'DATA SET CHANGE A+'/
0106
0107
0108
            (
0109
                        Define bits for MODEM/LINE STATUS from rxcsr register
0110
0111
0112
                                     dtr_status(0)
                                                             /'DATA TERMINAL READY = OFF+'/
                         Data
                                                             /'DATA TERMINAL READY = ON+'/
                         Data
                                     dtr_status(1)
```

DUP11

0114

C 15

16-Sep-1984 00:21:20 5-Sep-1984 13:53:59

VAX-11 FORTRAN V3.4-56

DISKSVMSMASTER: [ERF.SRC]DUP11.FOR: 1

```
D 15
DUP11
                                                                               16-Sep-1984 00:21:20
5-Sep-1984 13:53:59
                                                                                                            VAX-11 FORTRAN V3.4-56
                                                                                                            DISK$VMSMASTER: [ERF.SRC]DUP11.for; 1
0115
                    Data
                                                 /'REQUEST TO SEND = OFF+'/
                              rts_status(0)
0116
0117
                                                 /'REQUEST TO SEND = ON+'/
                    Data
                              rts_status(1)
0118
                                                 /'DATA SET READY = OFF+'/
                              dsr_status(0)
                    Data
                              dsr_status(1)
                                                 /'DATA SET READY = ON+'/
                    Data
0119
0120
0121
0123
0124
0125
0127
                    Data
                              car_status(0)
                                                 /'CARRIER DETECT = OFF+'/
                                                 /'CARRIER DETECT = ON+'/
                    Data
                              car_status(1)
                    Data
                                                 /'CLEAR TO SEND = OFF+'/
                              cts_status(0)
                    Data
                              cts_status(1)
                                                 /'CLEAR TO SEND = ON+'/
0128
0129
0130
                    Define text for bits in the RXDBUF register
0131
                             rxdbuf_1(8)
rxdbuf_1(9)
rxdbuf_1(10)
rxdbuf_2(12)
rxdbuf_3(14)
rxdbuf_3(15)
0132
0133
                                                 /'START OF RECEIVED MESSAGE+'/
                    Data
                                                 /'END OF RECEIVED MESSAGE+'/
                    Data
0134
                                                 /'RECEIVER ABORT+'/
                    Data
0135
                                                 /'RÉCÉIVÉR CRC ERROR+'/
                    Data
0136
0137
                                                 /'RECEIVER OVERRUN*'/
                    Data
                                                 /'RECEIVER ERROR*'/
                    Data
0138
0139
0140
0141
                    Define text for bits in the TXCSR register
0142
0144
                    Data
                             oper_mode(0)
                                                 /'FULL DUPLEX+'/
                             oper_mode()
oper_mode(1)
txcsr_1(4)
txcsr_2(6)
txcsr_2(7)
txcsr_2(8)
txcsr_2(9)
txcsr_3(15)
0145
                                                 /'HALF DUPLEX*'/
                    Data
0146
                                                 /'SEND+'/
                    Data
0147
                                                 /'TRANSMITTER INTERRUPT ENABLE*'/
                    Data
0148
                                                 /'TRANSMITTER DONE +'/
                    Data
0149
                                                 /'DEVICE RESET*'/
                    Data
0150
                                                 /'TRANSMITTER ACTIVE+'/
                    Data
0151
                                                 /'TRANSMITTER DATA LATE ERROR+'/
                    Data
0152
0153
0154
                    Define text for bits in the TXDBUF register
0156
0157
                             txdbuf_1(8)
txdbuf_1(9)
txdbuf_1(10)
0158
0159
                                                 /'TRANSMIT START OF MESSAGE+'/
                    Data
                                                 /'END OF TRANSMITTED MESSAGE+'/
                    Data
0160
                                                 /'TRANSMIT ABORT+'/
                    Data
0161
0162
0163
0164
                    Define text for bits in the PARCSR register
0165
0166
0167
                    Data
                              parcsr_1(9)
                                                 /'CRC INHIBIT+'/
0168
                    Data
                              protocol(0)
                                                 /'BIT ORIENTED PROTOCOL+'/
0169
                    Data
                             protocol(1)
                                                 /'BYTE ORIENTED PROTOCOL+'/
```

DUI

PP(

EN1

(

VAF

3

Page

```
E 15
                                                                                              16-Sep-1984 00:21:20
5-Sep-1984 13:53:59
DUP11
0172
0173
0174
0175
0176
0177
                       diagnostic_mode = .false.
                       if (lib$extzv(11,2,txcsr) .ne. 0) diagnostic mode = .true.
Ŏ178
0179
                       Decode receiver control and status register
0180
0181
0182
0183
                       Call LINCHK (lun,2)
Write (lun,20) rxcsr
format (/' ',T8,'RXCSR',T24,Z8.4)
0184
            20
0185
0186
                       if (.not. diagnostic_mode) then
0187
                       Call LINCHK (lun,1)
Write (lun,30) ('*',I=1,34)
Format ('',740,34A1)
0188
0189
0190
            30
0191
0192
                       Call LINCHK (lun,1)
                       Write (lun,40)
Format ('', T47, 'MODEM/LINE STATUS')
0193
0194
            40
0195
0196
                       field=LIB$EXTZV (1,1,rxcsr)
0197
0198
                       Call LINCHK (lun,1)
                       Write (lun,50) dtr_status(field)
Format (' ',T40,A<compress( (dtr_status(field))>)
0199
50
                       Field=LIB$EXTZV (2,1,rxcsr)
                       Call LINCHK (lun,1)
                       Write (lun,60) rts_status(field)
Format (' ',740,A<compress( (rts_status(field))>)
           60
                       Field=LIB$EXTZV (9,1,rxcsr)
                       Call LINCHK (lun,1)
Write (lun,70) dsr_status(field)
Format (' ,140,A<compressC (dsr_status(field))>)
            70
                       field=LIB$EXTZV (12,1,rxcsr)
                       Call LINCHK (lun,1)
Write (lun,80) car_status(field)
Format (' ',T40,A<compressC (car_status(field))>)
            80
                       Field=LIB$EXTZV (13,1,rxcsr)
                       Call LINCHK (lun,1)
Write (lun,90) cts_status(field)
format (' ',T40,A<compressC (cts_status(field))>)
```

DUI

ARI

1

LA

FUI

Page

VAX-11 FORTRAN V3.4-56 DISK\$VMSMASTER:[ERF.SRC]DUP11.FOR;1

```
Call LINCHK (lun,1)
Write (lun,100) ('*', I=1,34)
Format ('',740,34A1)
          100
                   Call DUTPUT (lun,rxcsr,rxcsr_1,0,0,0,'0')
                   Call OUTPUT (lun, rxcsr, rxcsr_2,3,3,8,'0')
                   Call OUTPUT (lun, rxcsr, rxcsr_3, 10, 10, 11, '0')
                   Call OUTPUT (lun, rxcsr, rxcsr_4, 14, 14, 15, '0')
                   endif
                   Decode receiver data buffer register
                   Call LINCHK (lun.1)
                   Write (lun,110) rxdbuf
format (* ,T8, RXDBUF*,T24,Z8.4)
          110
                   if (.not. diagnostic_mode) then
                   Field=L1B$EXTZV(0,8,rxdbuf)
                   Call LINCHK (lun,1)
                   Write (lun, 120) field format (' ', 140, 'RECEIVER DATA BUFFER = ',
          120
                   1 I<compress4 (field)>,'.')
0260
                   Call OUTPUT (lun,rxdbuf,rxdbuf_1,8,8,10,'0')
0261
0262
                   Call OUTPUT (lun,rxdbuf,rxdbuf_2,12,12,12,'0')
0263
0264
                   Call OUTPUT (lun,rxdbuf,rxdbuf_3,14,14,15,'0')
0265
                   endif
0266
0267
0268
                   Decode tranmitter control and status register
0269
0270
0271
                   Call LINCHK (lun,1)
0272
0273
                   Write (lun, 130) txcsr
format ('', T8, 'TXCSR', T24, Z8.4)
          130
0274
                   if (.not. diagnostic_mode) then
0276
0277
                   field=LIBSEXTZV(3,1,txcsr)
0278
0279
                   Call LINCHK (lun.1)
                   Write (lun, 140) oper mode(field) format (* ,140, OPERATION MODE = *,
0280
0281
          140
0282
0283
                   1 A<compressC (oper_mode(field))>)
0284
                   (all OUTPUT (lun,txcsr,txcsr_1,4,4,4,'0')
0285
```

```
G 15
 DUP11
                                                                                         16-Sep-1984 00:21:20
5-Sep-1984 13:53:59
                                                                                                                          VAX-11 FORTRAN V3.4-56
                                                                                                                          DISK$VMSMASTER: [ERF.SRC]DUP11.FOR:1
0286
0287
0288
0289
0291
0292
0293
0295
                       Call OUTPUT (lun,txcsr,txcsr_2,6,6,9,'0')
                       Call OUTPUT (lun,txcsr,txcsr_3,15,15,15,'0')
                       else
                       Call LINCHK (lun,1)
                      Write (lun, 150)
Format (', 740, 'DIAGNOSTIC MODE')
           150
                       endif
0296
0297
0298
0299
0300
                       Decode transmitter data buffer register
                       Call LINCHK (lun,1)
                      Write (lun, 160) txdbuf
format (' ', T8, 'TXDBUF', T24, Z8.4)
0301
0302
           160
0304
                       if (.not. diagnostic_mode) then
0305
0306
                       Field=LIB$EXTZV(0,8,txdbuf)
0307
                      Call LINCHK (lun,1)
Write (lun,170) field
format ('',140,'TRANSMITTER DATA BUFFER = ',
1 I<compress4 (field)>,'.')
0308
0309
0310
           170
0311
0312
                      Call OUTPUT (lun,txdbuf,txdbuf_1,8,8,10,'0')
0314
                      endif
0315
0316
0317
0318
0319
0321
0322
0323
0324
0326
0327
0328
                      Decode and output PARCSR register contents
                      Call LINCHK (lun,1)
                      Write (lun, 180) parcsr
format (' ', 18, 'PARCSR', 124, 28.4,/,
1 140,'** LOADED WITH **')
           180
                      if (.not. diagnostic_mode) then
0329
0330
                      Call OUTPUT (lun,parcsr,parcsr_1,9,9,9,'0')
0331
                       Temp=LIB$EXTZV (12,1,parcsr)
0332
                       field=LIBSEXTZV (15,1,parcsr)
0334
                      (all LINCHK (lun,1)
Write (lun,190) protocol(field)
format (' ,140, A<compress( (protocol(field))> )
0335
0336
0337
           190
0338
0339
                       Call LINCHK (lun,1)
0340
                       If (field .eq. 0 .and. temp .eq.0) then
0341
0342
                       Write (lun, 200)
```

DUI

03

PR(

EN'

(

VAI

Al

Page

	H 15 16-Sep-1984 00:21:20	Page	
200	Format (' ',T40,'OPERATING AS A PRIMARY STATION')		
	Else if (field .eq. 0 .and. temp .eq. 1) then		
210	Write (lun,210) format (' ',T40,'OPERATING AS A SECONDARY STATION')		
	Addrs=LIB\$EXTZV (0,8,parcsr)		
220	<pre>Call LINCHK (lun,1) Write (lun,220) addrs format (' ',T40,'SECONDARY STATION ADDRESS = ', 1 I<compress4 (addrs)="">,'.')</compress4></pre>		
	Else		
	Char=LIB\$EXTZV (0,8,parcsr)		
230	Write (lun,230) char format (' ',140,'EXPECTED SYNC CHAR = ', 1 I <compress4 (char)="">,'.')</compress4>		
	Endif endif		
	Return		
	END		

DU

FU

co

CO

## PROGRAM SECTIONS

Name	Bytes	Attributes
O SCODE 1 SPDATA 2 SLOCAL	2023 592 2024	PIC CON REL LCL SHR EXE RD NOWRT LONG PIC CON REL LCL SHR NOEXE RD NOWRT LONG PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	4639	

# ENTRY POINTS

Address Type Name 0-00000000 DUP11

## VARIABLES

Address Type	Name	Address Type	Name
2-0000039C 1*4	ADDRS DIAGNOSTIC_MODE I PARCSR RXDBUF TXCSR	2-000003A0 I+4	CHAR
2-0000399 L*1		2-000003A4 I+4	FIELD
2-00003AC I*4		AP-00000004a L+1	LUN
AP-00000008a I*4		AP-0000000Ca I+4	RXCSR
AP-00000010a I*4		2-00003A8 I+4	TEMP
AP-00000014a I*4		AP-00000018a I+4	TXDBUF

# ARRAYS

Address	Type	Name	Bytes	Dimensions
2-0000347 2-0000371 2-000031D 2-00002BD 2-000002E 2-000002F1 2-000002F1 2-0000058 2-0000058 2-0000124 2-0000124 2-0000148 2-0000100 2-00001D0 2-0000260	CHAR CHAR CHAR CHAR CHAR CHAR	CAR_STATUS CTS_STATUS DSR_STATUS DTR_STATUS OPER_MODE PARCSR_1 PROTOCOL RTS_STATUS RXCSR_1 RXCSR_2 RXCSR_3 RXCSR_4 RXDBUF_1 RXDBUF_3 TXCSR_2 TXCSR_3 TXCSR_3 TXCSR_3 TXCSR_3	420 422 442 1444 1548 1548 179 1168 1168	(0:1) (0:1) (0:1) (0:1) (0:1) (0:1) (0:1) (0:0) (3:8) (10:11) (14:15) (8:10) (14:15) (4:4) (6:9) (15:15) (8:10)

J 15

VAX-11 FORTRAN V3.4-56 DISK\$VMSMASTER:[ERF.SRC]DUP11.FOR;1

ı	A	٥	E	ı	c
L	_	0	C	L	3

Address	Label	Address	Label	Address	Label	Address	Label	Address	Label	Address	Label
1-0000038 1-00000095 1-0000104 1-000001Af	20' 80' 140' 200'	1-0000004E 1-000000A1 1-00000123 1-000001D5	30' 90' 150' 210'	1-0000058 1-000000AD 1-000013A 1-000001FD	40' 100' 160' 220'	1-00000071 1-000000B7 1-0000014D 1-0000022A	50' 110' 170' 230'	1-0000007D 1-000000CA 1-00000178	60' 120' 180'	1-00000089 1-000000F2 1-000001A3	70' 130' 190'

#### FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	Type	Name	Type	Name	Type	Name	Type	Name
1=4	COMPRESS4	1+4	COMPRESSO	1+4	LIBSEXTZV		LINCHK		OUTPUT

#### COMMAND QUALIFIERS

FORTRAN /LIS=LIS\$:DUP11/OBJ=OBJ\$:DUP11 MSRC\$:DUP11

```
/CHECK=(NOBOUNDS,OVERFLOW,NOUNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP)
/F77 /NOG_FLOATING /14 /OPTIMIZE /WARNINGS /NOD_LINES /NOCROSS_REFERENCE /NOMACHINE_CODE /CONTINUATIONS=19
```

#### COMPILATION STATISTICS

5.30 seconds 13.85 seconds Run Time: Elapsed Time: Page faults:

201 200 pages Dynamic Memory:

# 0147 AH-BT13A-SE VAX/VMS V4.0

# DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY

1 min

	Fig. 19 (1997) Annual Control of	The second secon	CLASSIFY LIS	Great sustain		Control of the contro	Specification of the second se	2 131 and 191	Villed 191	E Tilenen	Garas, mannes,	General August	B. THE SECOND	TODACA CALL  TODAC	Great Mattering of The Control of Th
E III III II	Anna Gana Anna Anna Anna Anna Anna Anna	The state of the s	TOTAL CONTROL OF THE PROPERTY	The second	B Jilled 198	THE SAME	2 111 ran 191	RIPE BILLS			the second	The State of		State former.	
	The second secon	in Tinisaisa	Section 1	SEC. U.T.	Mark 16	I III BE	Man space to make make make make make make make make	DR250 LIS	SEE 122	Wester Australia	Charte manage.	dans stands		Links Market Links	Interior
The second secon	Section Control of Con	Mathematical Mathe	B THE SALE	CSTRING . LIS	DHEADS LIS	Section 1.	BESSEL	Market Administration (Market Administration	DR780 LIS	2 THE RESIDENCE	Harris Andrews	Mentional Park Manager 198 (1984) (19	these manner.	Green market	the like III
	Market Ma		Total Residence Control Reside	WE THE PROPERTY OF THE PROPERT	Marin	Company Compan	DR11W LIS	Hater Admira	Shorts manuful and the same of	Manual State of State	Service matrices		B SHIP AND THE	CONTRACTOR OF THE PROPERTY OF	there agents——
To the national section of the secti	Section And Section Assessment Control of the Contr	THE STATE OF THE S	COMPRESS LIS	With the second	than many	** *** *** *** *** *** *** *** *** ***	March	Sign Matter Address.	the constant of the constant o	DTAILS LIS	Martin Ma		Same and the same	General Section Control of Contro	them associated from
For the Comment of th	The second secon	Timed yet	TECHNOLOGY PARTY STATES OF THE PARTY STATES OF	DECODECC LIS	Company Agents	Manufacture Manufa	I L II WAR CICIO- CICIO	GOTON AMERICAN	I DE DESCRIPTION DE LA CONTRACTION DEL CONTRACTION DE LA CONTRACTI	Gentle Marie Communication of the Communication of	Grand Address	Control Section Control Sectin Control Section Control Section Control Section Control Section	State Cam	Service Administration of the Control of the Contro	MINE WINE
	The state of the s	William Removed Proportion State Part		TOTAL CONTROL OF THE PROPERTY	MATTER MATTER AND	Section 1997 Annual Property of the Control of the	The second secon	NATIONAL MARKET PROPERTY OF THE PROPERTY OF TH	Maria nameta  Simple and nameta  Simpl	Garter manters.	The state of the s	The second secon	DUMPREG LIS	Time agence	DUTUDRIVR LIS
	TOTAL STATE AND THE STATE AND	CALCMAP LIS	Ment dahara		Matte Autorities	And the second s	Martin Ma	Marine Authors  - VESTINA  - VEST	Grand Authority			Sign and American Control of the Con	Secretarian	ms ms ms ms ms ms ms	The second secon
The second secon		CONTROL OF THE PROPERTY OF THE			P IIII da 191			- WEARDOOM, ASSESSMENT PROPERTY OF THE PROPERT			Unite automate.	Control States and tree	Santa particular	DUP3271 LIS	The state of the s
SCHOOLS MARKET M			980000 60s. 12"		Section same	3 illian ian	THE DR	- 10 20 65 - 10 10 10 10 10 10 10 10 10 10 10 10 10					Service supervised by the service supervised	Charles management (Charles management (Charle	When the second
		Service Services	- CRYPTK LIS		DQDISKS LIS			- Marin and Mari						The same and the s	The second secon
Control Contro		9660953 162	Management Comments of the Com		TOTAL TOTAL STATE OF THE STATE								DUP11		